Claim Amendments

1.   (Currently amended). A method for allowing objects in a first programming language to communicate with objects in a second programming language, comprising:

   a) receiving metadata information from a server running said second programming language on a client running said first programming language;

   b) generating proxies for said first programming language from said metadata information, using a development tool for said first programming language, wherein said proxies are generated by a one-to-one mapping of classes from said second programming language to said first programming language <u>and said proxy is customized for the client's intended purpose</u>; and

   c) implementing said proxies on said client, wherein said method is provided solely in said first programming language and said client does not require any components from said second programming language.

2.   (Original). The method according to claim **1**, including an additional step d) using said proxies to enable bi-directional communication between said client and said server.

3.   (Previously amended). The method according to claim **1**, wherein said first programming language is a JAVA cross platform programming language and said second programming language is common language runtime (CLR).

4.   (Previously amended). The method according to claim **1**, wherein said first programming language is a .Net Remoting

programming language and said second programming language is a JAVA cross platform programming language.

5.   (Original). The method according to claim **1**, wherein said client and said server communicate using SOAP formatted messages.

6.   (Original). The method according to claim **1**, wherein said client and said server communicate using binary formatted messages.

7.   (Original). The method according to claim **1**, including the additional step of passing said proxies to a runtime tool using said first programming language.

8.   (Original). The method according to claim **7**, wherein said runtime tool is capable of operating independently of said development tool.

9.   (Currently amended). A system enabling bi-directional communication using .Net Remoting protocol between JAVA objects in a JAVA virtual machine environment and .Net assemblies objects in a common language runtime (CLR) environment, comprising:

   (a)   a computer network;

   (b)   a JVM computer having random access memory (RAM) and at least one of hard disk storage memory (HDS) and solid state storage memory (SSSM), said computer having a JAVA Virtual Machine (JVM) environment and JAVA objects in one of said HDS and SSSM, said JVM computer coupled to said computer network;

   (c)   a CLR computer having random access memory (RAM) and at least one of hard disk storage memory (HDS) and solid state storage memory (SSSM), said computer having a

CLR environment and .Net assemblies in one of said HDS
and SSSM, said CLR computer coupled to said network;

(d)   a JAVA development computer with a RAM, and one of
HDS and SSSM, said JAVA development computer having a JVM
environment and a JAVA-based tool in one of said HDS or
SSSM, said JAVA development computer coupled to said
network, wherein said JAVA-based tool is used, during
development, to select .Net assemblies running on CLR
computers), wherein

        (i)   said JAVA-based tool being used during
        development to select .Net assemblies running on
        said CLR computer on said computer network and to
        generate a corresponding set of JAVA proxies
        customized according to their intended purpose;
        and

        (ii) said JAVA proxies are copied onto said JVM
        computer and are operative to allow said JAVA
        objects to communicate with selected .Net
        assemblies on said CLR computer;

(e)   a CLR development computer having memory
comprising RAM, and one of HDS and SSSM, and having a CLR
environment in said memory, said CLR development computer
coupled to said computer network, and having a CLR-based
tool in said memory operative during development to
select specified JAVA objects on said JVM computer over
said computer network and to generate a corresponding set
of CLR proxies, wherein said CLR proxies are copied onto
said CLR computer and are operative to allow said CLR
objects to communicate with said specified JAVA objects

on said CVM computer, <u>said CLR proxies being customized</u> <u>according to their intended purpose</u>.

10.   (Previously amended).   The system of claim **9**, further comprising a JAVA cross platform programming language based runtime tool stored on said one computer for handling said JAVA proxies and said .Net proxies.

11.   (Previously amended).   The system of claim **10**, wherein said JAVA cross platform programming language based runtime tool is capable of operating independently of said JAVA cross platform programming language based tools for generating JAVA and .Net proxies.